<table>
<thead>
<tr>
<th>Week</th>
<th>Dates</th>
<th>Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1/13 - 1/17</td>
<td>Introductory Fundamentals, Proof</td>
</tr>
<tr>
<td>2</td>
<td>1/20 - 1/24</td>
<td>Finite Automata, breeze</td>
</tr>
<tr>
<td>3</td>
<td>1/27 - 1/31</td>
<td>Non determinism, NFA - DFA equivalence, Regular Expressions</td>
</tr>
<tr>
<td>4</td>
<td>2/3 - 2/17</td>
<td>Regular Expressions &amp; NFA</td>
</tr>
<tr>
<td>5</td>
<td>2/10 - 2/14</td>
<td>Pumping Lemma (2/10), Apply PL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2/14 (CP6)</td>
</tr>
<tr>
<td>Week</td>
<td>Dates</td>
<td>Notes</td>
</tr>
<tr>
<td>--------</td>
<td>-----------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>Week 6</td>
<td>2/12</td>
<td>Holiday</td>
</tr>
<tr>
<td></td>
<td>2/14</td>
<td>CPG, Ambiguity, CNF</td>
</tr>
<tr>
<td></td>
<td>2/14</td>
<td>PDA</td>
</tr>
<tr>
<td>Week 7</td>
<td>2/24</td>
<td>PDA (2)</td>
</tr>
<tr>
<td></td>
<td>2/26</td>
<td>Review for mid term</td>
</tr>
<tr>
<td></td>
<td>2/28</td>
<td>Mid Term 1</td>
</tr>
<tr>
<td>Week 8</td>
<td>3/2</td>
<td>Today</td>
</tr>
<tr>
<td></td>
<td>3/4</td>
<td>Prof. Dang (See 2.3)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Non Context Free Language</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pumping Lemma</td>
</tr>
<tr>
<td></td>
<td>3/6</td>
<td>Part 2 of Dang</td>
</tr>
<tr>
<td></td>
<td></td>
<td>James CollecHW 5</td>
</tr>
<tr>
<td>Week 9</td>
<td>3/9</td>
<td>I'm gone (Travel)</td>
</tr>
<tr>
<td></td>
<td>3/13</td>
<td>I'm back (2.4)</td>
</tr>
</tbody>
</table>

March 16-20 Spring Break
Theorem

A language is Context-free iff

Some pushdown Automaton recognizes it.

⇒ Part 1

If a language is Context-free, then some pushdown Automaton recognizes it.
Let $A$ be a CFL.

We show how to convert a CF for $A$ from an $G$ into an equivalent PDA, called $P$.
Informal description of P.

1. Place the marker symbol # and the start variable in the stack.

2. Repeat the following forever
   (a) If the top of the stack is a variable symbol A, nondeterministically select one of the rules for A and substitute A by the string on the RHS of the rule.
b) If the top of the stack is terminated by an a, read the next symbol from the input and compare with a. If they match, repeat.

If they do not match, reject on this branch of non-determinism.

c) If the top of the stack is b, enter the accept state. Only do accepts input if it has been read.
Proof

Let $P = (Q, \Sigma, \Gamma, S, q_0, \delta, \delta', F)$. 

To make construction of $P$ simpler, we use a short-hand notation for transition function.

This short-hand gives a way to write an entire string on the tape in one step.

We can simulate this action by introducing additional states, to write the string one symbol at a time.
Let $q$ and $r$ be states of the PDA, and let $a$ be in $\Sigma$
let $s$ be in $\Gamma$

Say we want the PDA to go from $q$ to $r$ when $a$ reads $a$ and pops $s$.
Furthermore, we want the PDA to push the entire string $u = u_1 \ldots u_k$ into the stack at the same time.

We can implement this action by introducing new states $q_1, \ldots, q_{k+1}$
and transition function as follows.
$S(q, a, s)$ to contain $(q_2, a, \varepsilon)$,
$S(q_2, a, s) = \{(q_2, a, \varepsilon)\}$,
$S(q_2, \varepsilon, s) = \{(q_3, a, \varepsilon)\}$,
$S(q_3, \varepsilon, s) = \{(q_4, a, \varepsilon)\}$.

We use the notation
$(r, u) \in S(q, a, s)$ to mean
that when $q$ is the state of the automaton
A is the next symbol, and $S$ is
the symbol on the top of the stack,
the PDA may read a $a$ and pop the $s$,
then push the string $u$ onto the stack
and go to state $r$. 

In picture,
The states of \( P \) are \( Q = \{ \text{Start}, \text{Stop}, \text{Accept} \} \cup E \),

where \( E \) is the set of states we need for implementing the namespace.

The start state is \( \text{Start} \).

The only accept state is \( \text{Accept} \).
The transition function is defined as follows.

We begin by initializing the state to contain symbols $a$ and $b$, implementing Step 1 of the informal description:

$$S(\{a, b\}, 3) = \{(3), (3, 4), (3, 5), (3, 6), (3, 7)\}.$$ 

Then we will do Step 2 of the informal description.
First case (Case a), where the top of the stack contains a variable.

Let $S(S_{\text{loop}}, \varepsilon, A) = \{S_{\text{loop}}, \mathit{w}\}$

where

$A \rightarrow \mathit{w}$ is a rule in $\mathit{R_2}$.

Second case (b), where the top of the stack contains a terminal.

Let $S(S_{\text{loop}}, a, a) = \{S_{\text{loop}}, \varepsilon, (3)\}$.

Finally, case c (top of stack is empty)

Let $S(S_{\text{loop}}, \varepsilon, \#) = \{S_{\text{accept}}, \varepsilon, 3\}$.
Example

Construct a PDA $P_1$ from the following CFG $G$.

$$S \rightarrow aTb \mid b$$

$$T \rightarrow Ta \mid \varepsilon$$

The transition function is shown in the following diagram.